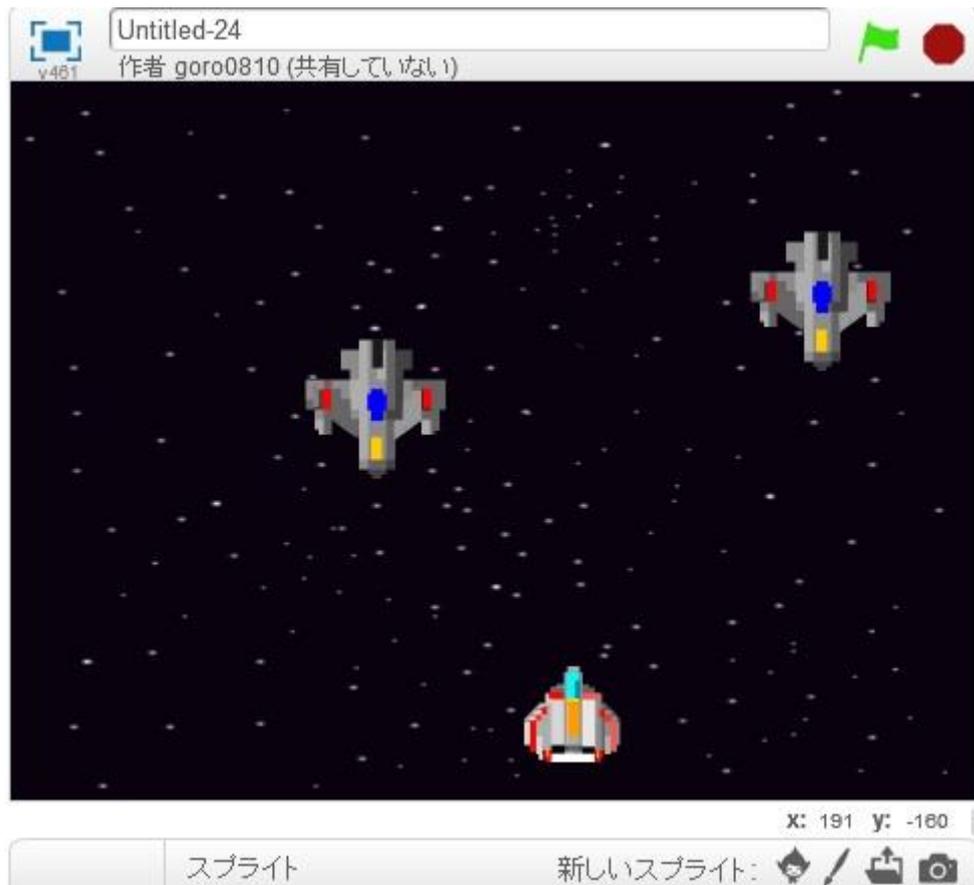


スクラッチを使って シューティング ゲームをつくらう



数楽問題提供サイト数楽

<http://www.mathtext.info/>

プログラミングとは

プログラミングとは、決まったことをコンピュータやロボットにしてもらうために書く、マニュアル(作業手順書)のことです。

身近なもので例をあげると、自動販売機がそうです。お金を入れて、ボタンを押すと希望の飲み物が出てくる。おつりが必要であれば、適切な金額のおつりが出てきます。そのようにプログラムされていなければ、だれかほかの人が代わりにそれをしなければなりません。

コンピュータやロボットにこんなことしてほしいなと思ったら、そのようにしてもらうように、マニュアルをつくっていきます。そうすることで、コンピュータやロボットはそのマニュアル通り動くことになります。このようなマニュアルをつくることをプログラミングといいます。

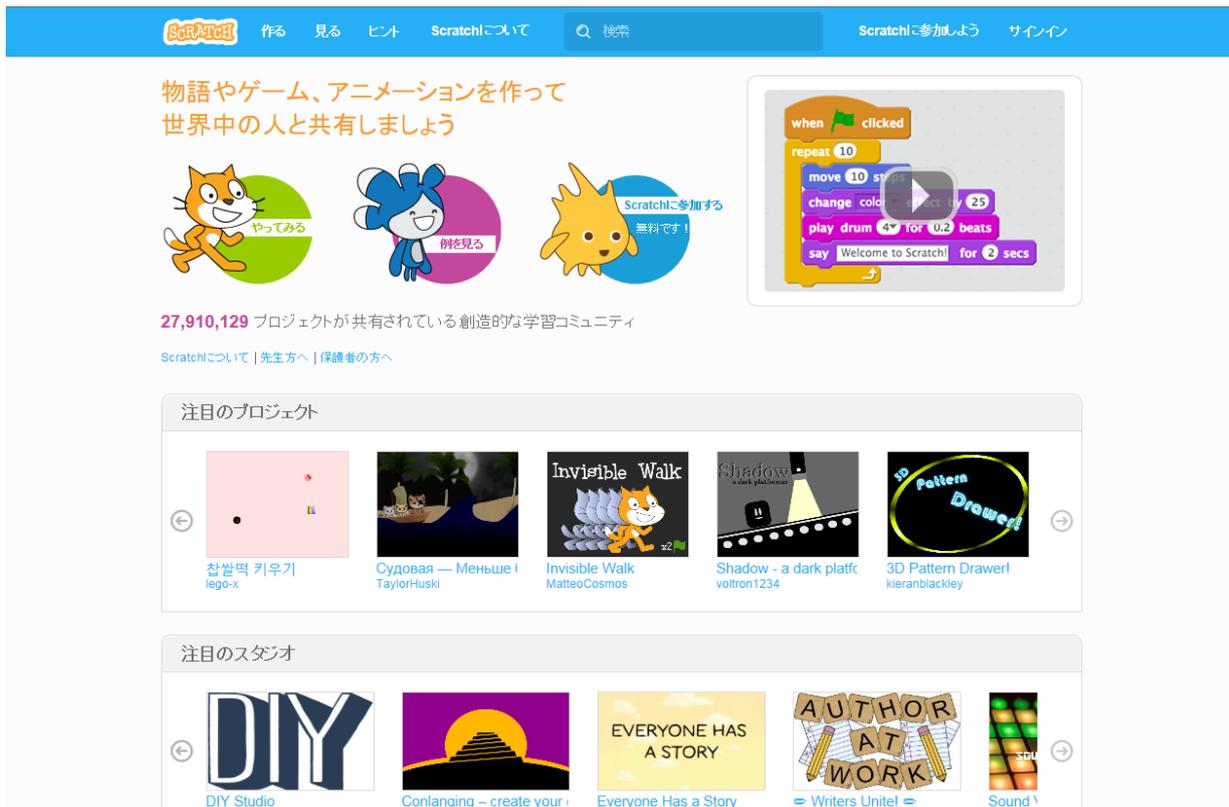
プログラミングができれば、人の役にものすごく役に立ちます。切符の券売機、インターネットオークション、スマホアプリ、Googleなどの検索画面など身近なものがほとんど、プログラミングで、できています。

プログラムが動く仕組み

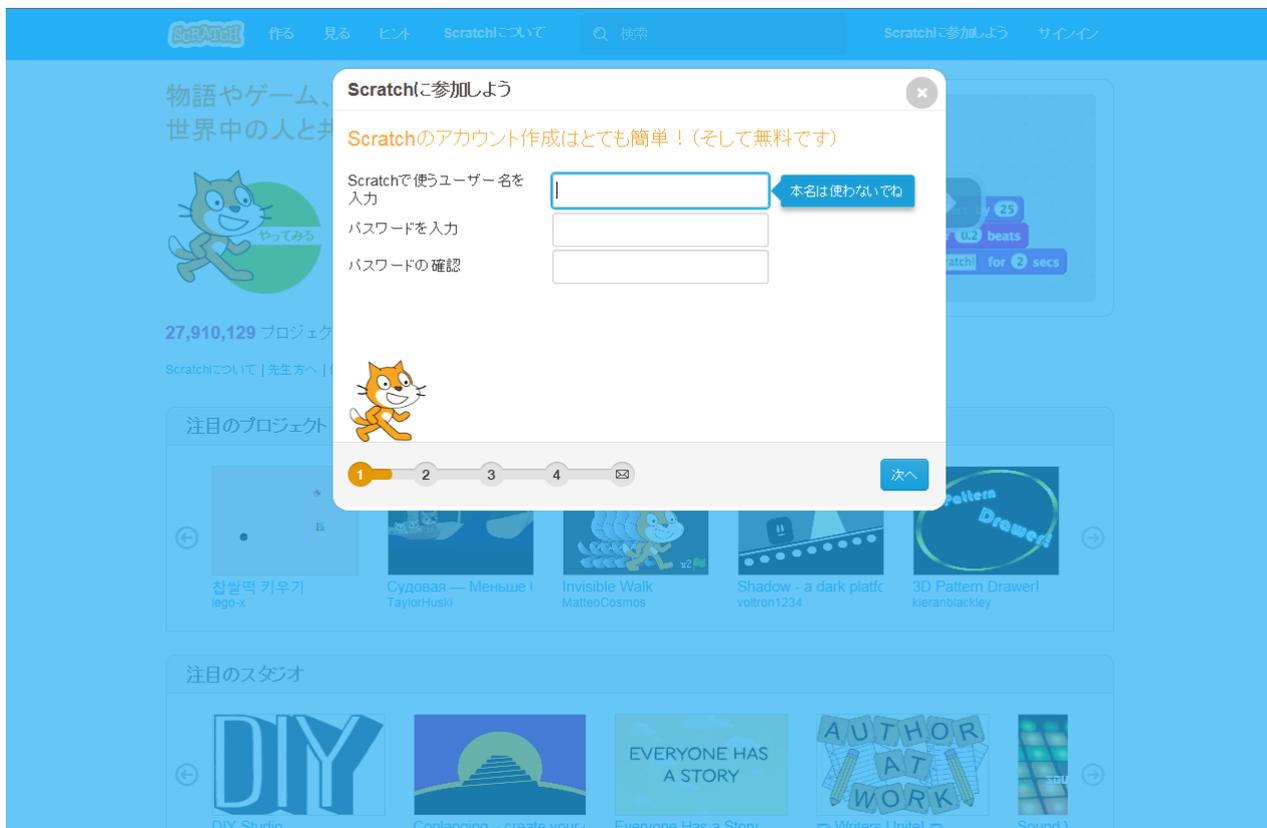
コンピュータなどの電子機械は信号があるかないかで動きます。専門的には2進数というもので動くことになります。どのように動くのか決めたのがマニュアルですが、コンピュータにはそのマニュアルは読めません。ですから、つくったマニュアルをコンピュータが読めるように電気信号に変換します。それがソフトウェアというものです。マニュアルをソフトウェアで電気信号に変えることで、コンピュータの頭脳であるCPUを動かして、期待する動作を得ることができます。

*プログラムが正確に書かれていないと、コンピュータは正常に動きません。場合によっては、ピクリともしません。そのあたりをわかっておかないと、プログラミングが楽しくなくなります。今回取り上げるスクラッチには、たくさんのサンプルプログラミングの本が出ておりますから、やってみたいと思った人は、本屋さんでご購入ください。

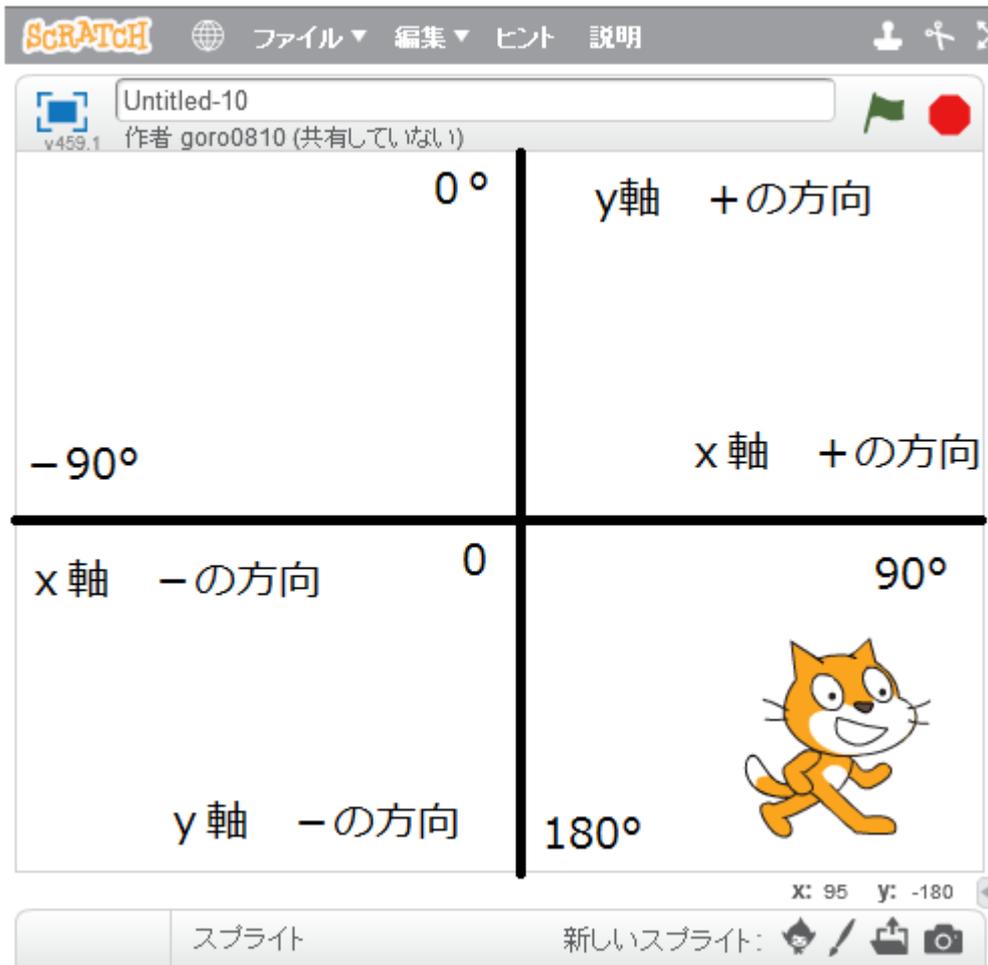
スクラッチを始めるには、スクラッチ MIT と検索をかけて、以下の画面に行きます。



サインインから、初期設定を行います。ただし、保護者の方の許可は得ましょう。各家にはそれぞれ家のルールがありますからね。



スクラッチでプログラミングしていくときは、画面が図のように設定されていますので、注意してつくっていきましょう。

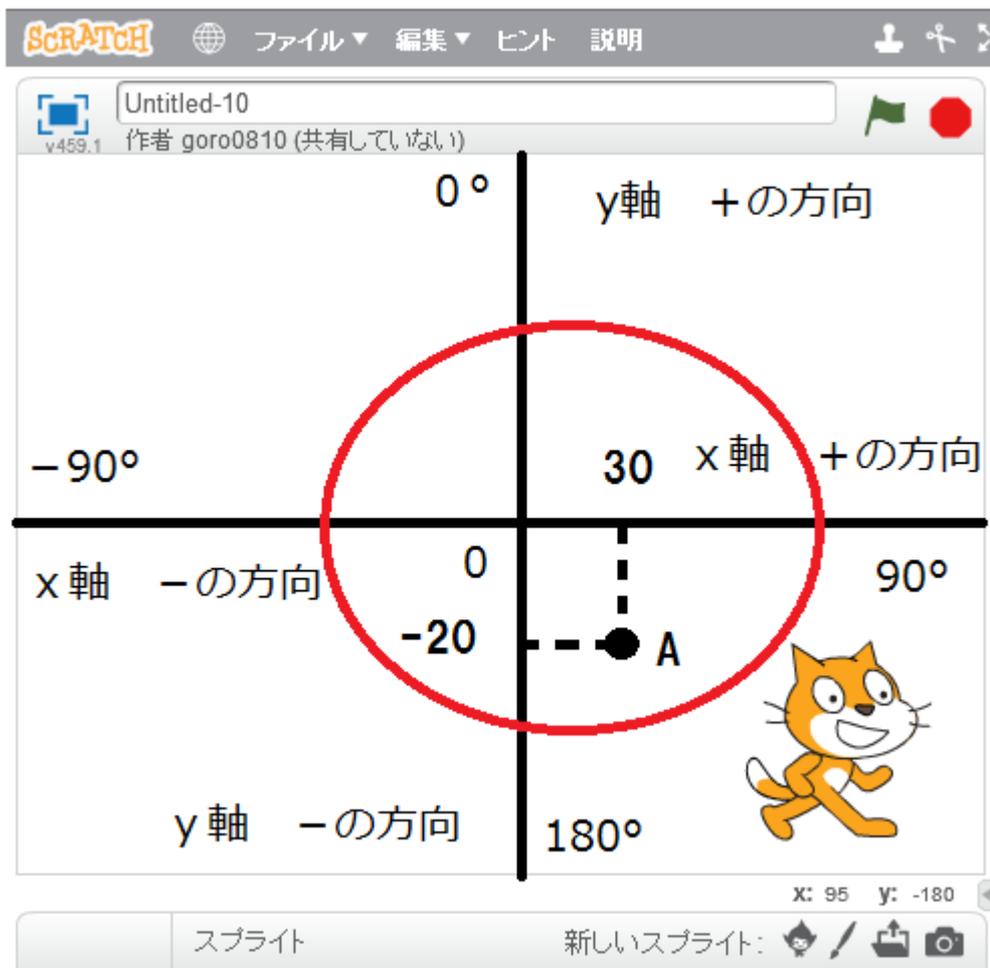


画面の中心が0というところです。0から横に伸びた線をx軸、0から縦に伸びた線をy軸といいます。

0はxもyも0のところなので、(x、y)の順番に(0、0)というように書きます。これを座標といいます。

スクラッチの画面は
横方向(x軸)は-240~+240の範囲
縦方向(y軸)は-180~+180の範囲
で構成されています。

例えば下の図で、Aはxが30、yが-20のところにありますから、Aの位置を(30、-20)という座標で表します。キャラクターをAに表示させたいければ、xが30、yが-20のところにキャラクターを表示するプログラムを書けば、表示することができます。キャラクターなどをコントロールするために、座標は必要になります。x(横方向)は-240~240の範囲、Y(縦方向)は-180~180の範囲で座標は設定可能です。



キャラクターや背景を設定する場合に使うボタンを○で囲っておきます。



 このボタンは自分で作ったものをアップロードするときに使います。

 このボタンは自分でキャラクターや背景をつくっていくときに使います。

まずゲームをつくるのには設計図が必要です。ちょうど家を建てるのに設計図が必要なのに同じように。

ですから、今回はシューティングゲームということで、自分の飛行機があってミサイルが発射されて、ミサイルが敵に当たれば敵が消える。ゲームオーバーの設定は敵が自分の飛行機に当たったらゲームオーバーとします。細かな設定は作りながら考えていいでしょう。実際つくってみたほうが分かることもありますから。

ゲームの設計(内容)は途中でゲームの設計を変えてもいいですし、途中で失敗してもいいです。とにかく上達するには失敗をくり返し、その分、上達していくことです。失敗してすることが、プログラミングでは許されています。

こうやって完成イメージをつかんでおきましょう。

ではつくっていきましょう。

新しいキャラクター(スプライト)を追加のボタンを押してキャラクターを取り込みます。
自分の飛行機、敵の飛行機、ミサイルです。



今回のプログラムでは
自分の飛行機は `playing111`
ミサイルは `bomb`
敵の飛行機は `atchr1`
という英数字で扱います。

猫のキャラクターはいらないので、削除しましょう。

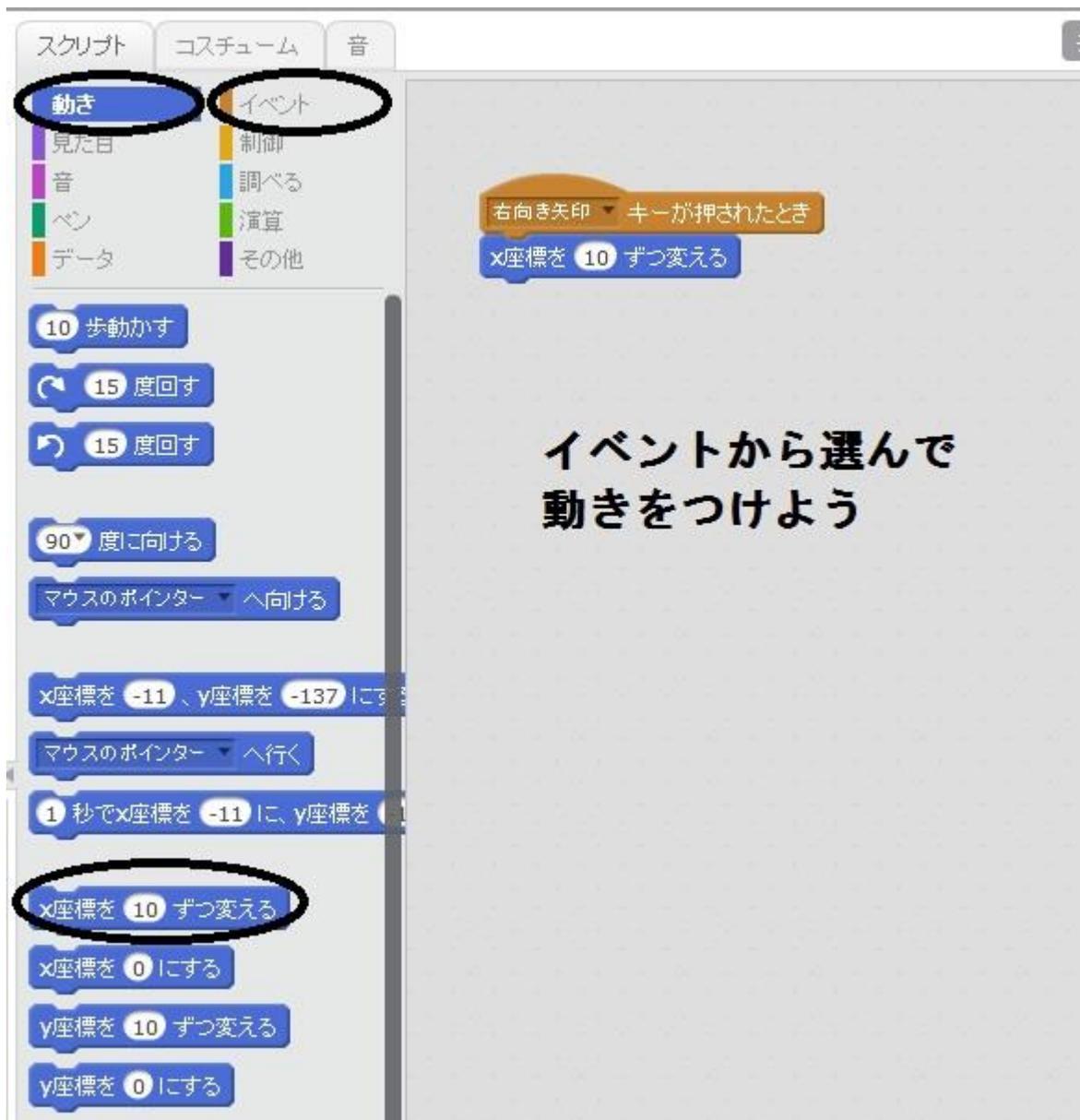
猫をクリックして、コスチュームを選ぶとこのような画面になるから○で囲ってある×印で削除していきましょう。消えないときは、猫の上で右クリック→削除でも消せることができます。



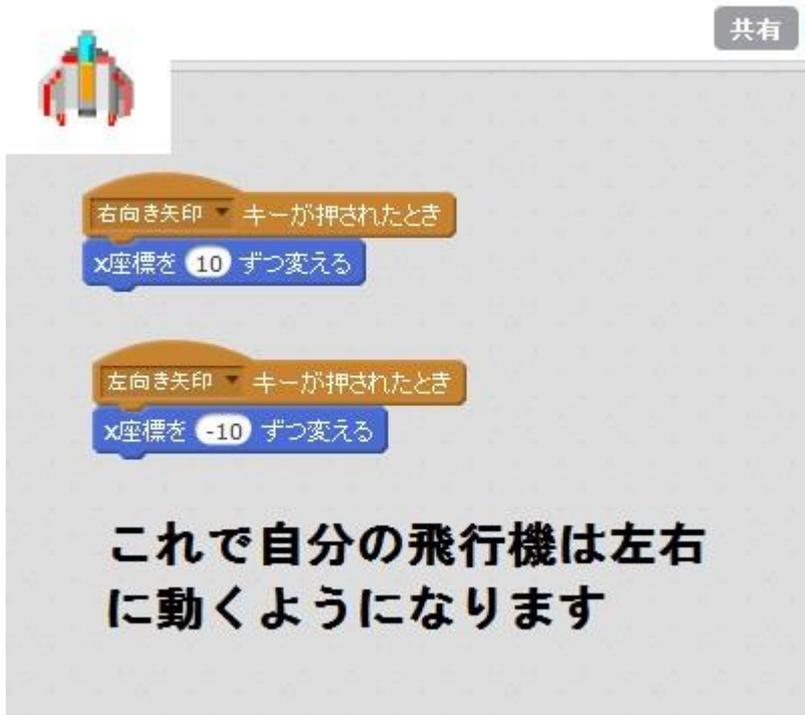
では早速プログラミング



自分の飛行機を選んだらスクリプトというタブを選びましょう。
右向き矢印キーが押されたときというイベントを置いてその下に動きから x 座標を 10 ずつ
変えるを追加しましょう。10 と言うのは右側に 10 動かすという意味です。



同じように左向き矢印キーが押されたときを作ります。このとき動きは-10にしましょう。
-10は左に10動かすという意味です。



10を20にすると自分の飛行機の動きは速くなります。

10より小さい数字だとゆっくり動きます。

上下に動かしたいときは上向き矢印、下向き矢印をつくりプログラミングすればできますが、今回は省略します。

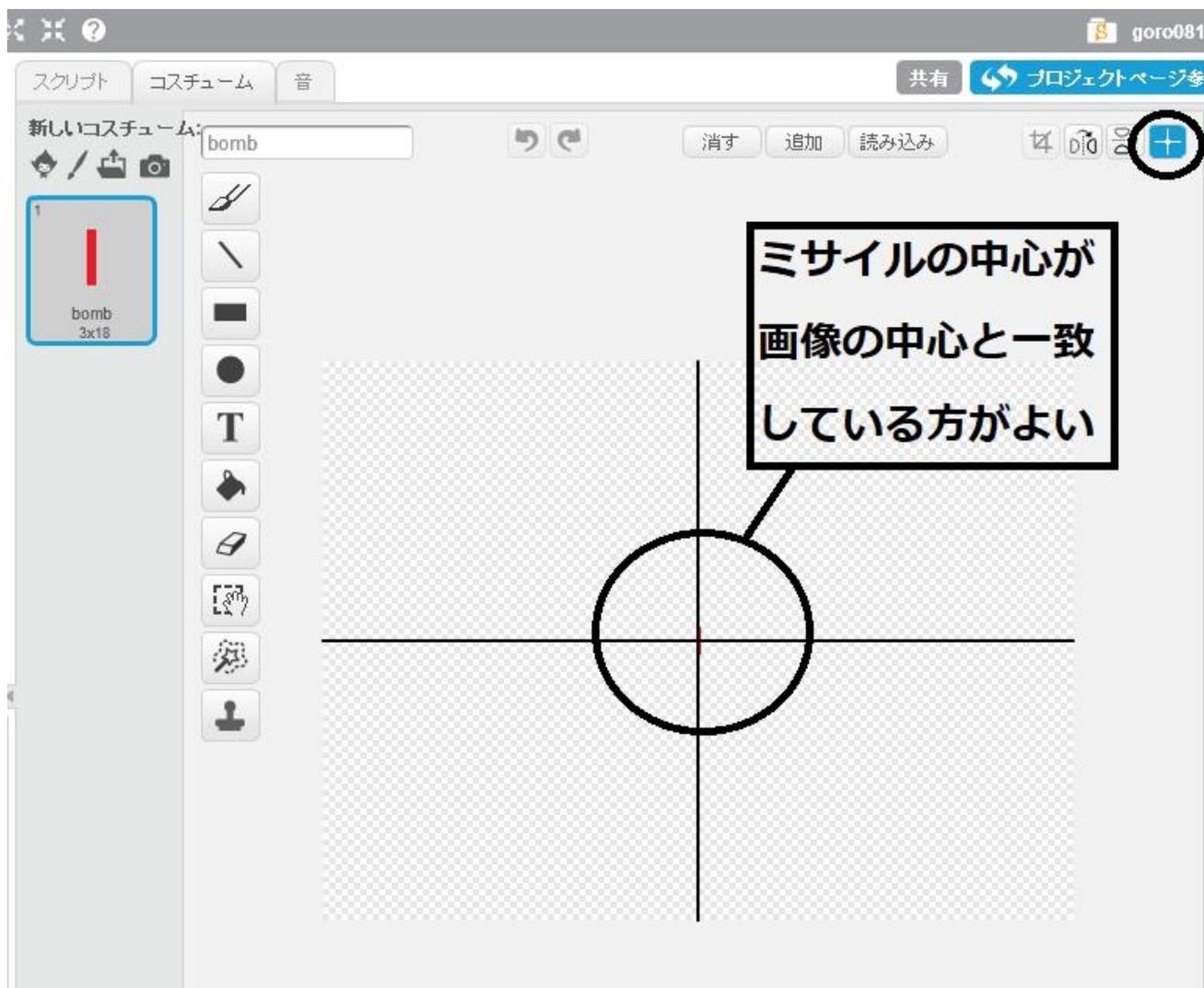
次にミサイルの動きを決めましょう。

スペースキーをミサイルの発射ボタンとして、スペースキーが押されたらミサイルが出るようにしましょう。



このとき注意が必要なのは、**ミサイルの中心が画像の中心と一致していることが好ましいです。**一致していることで、プログラムしやすくなります。一致していないときは+を動かして、画像の中心と+の中心を合わせてやりましょう。そうすればOKです。

もちろんこのことは、自分の飛行機、敵の飛行機にも言えることです。中心の位置を確認しておきましょう。中心がずれていたら、修正しておきましょう。



ミサイルを選択したときのスプライトに下のようプログラミングしましょう。

スクリーンショットのScratchプログラミング環境。左側のパレットには「動き」「見た目」「音」「ペン」「データ」のタブがあり、「イベント」タブが選択されている。右側のスクリプトエリアには以下のブロックが組み込まれている:

- スペースキーが押されたとき
- playimg111へ行く
- 端に触れたまで繰り返す
- y座標を10ずつ変える

プログラムの意味

イベントでスペースキーが押されたら

↓

自分の飛行機へ行く

↓

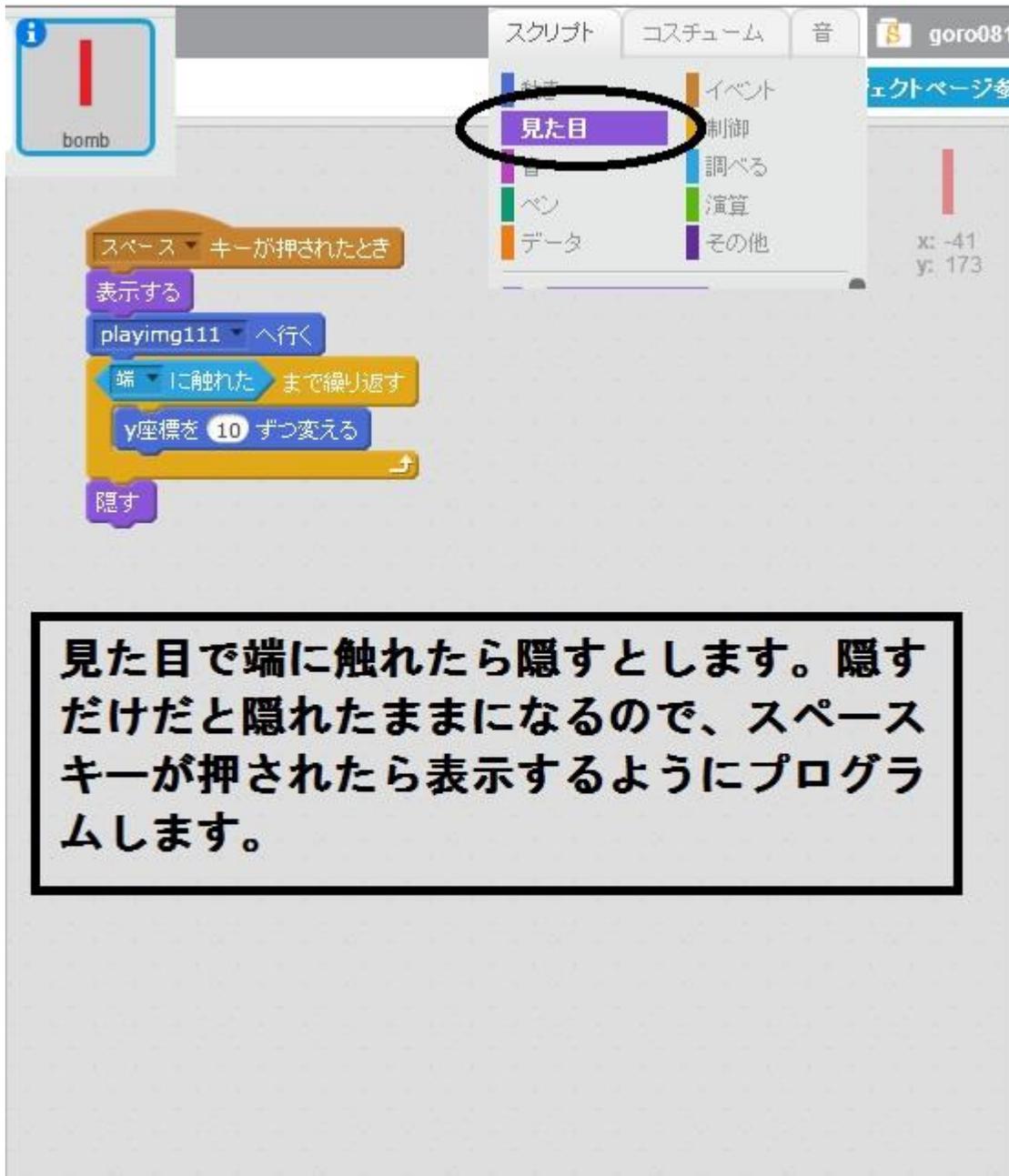
端に触れるまで上に10ずつすすめる

どうです？ミサイルが出ましたか？
でもここで問題です。



端に行ったらミサイルを消したいので、次のように変更します。

一番下に見た目から「隠す」を選択します。ただ、隠すだけなら、隠した後は隠れたままなので、スペースキーが押されるたびに見た目から「表示する」も追加しておきます。



The screenshot shows the Scratch code editor interface. On the left, a 'bomb' object is visible. The main workspace contains a script starting with 'when space key is pressed', followed by 'show', 'go to playimg111', a 'repeat until end reached' loop containing 'change y coordinate by 10', and finally 'hide'. A dropdown menu is open over the 'show' block, with '見た目' (Appearance) selected and circled. The menu also shows '制御' (Control), '調べる' (Debug), '演算' (Operators), and 'その他' (Miscellaneous). On the right, the object's coordinates are shown as x: -41, y: 173. At the bottom, a text box explains the logic of showing and hiding the object.

見ただ目で端に触れたら隠すとします。隠すだけだと隠れたままになるので、スペースキーが押されたら表示するようにプログラムします。

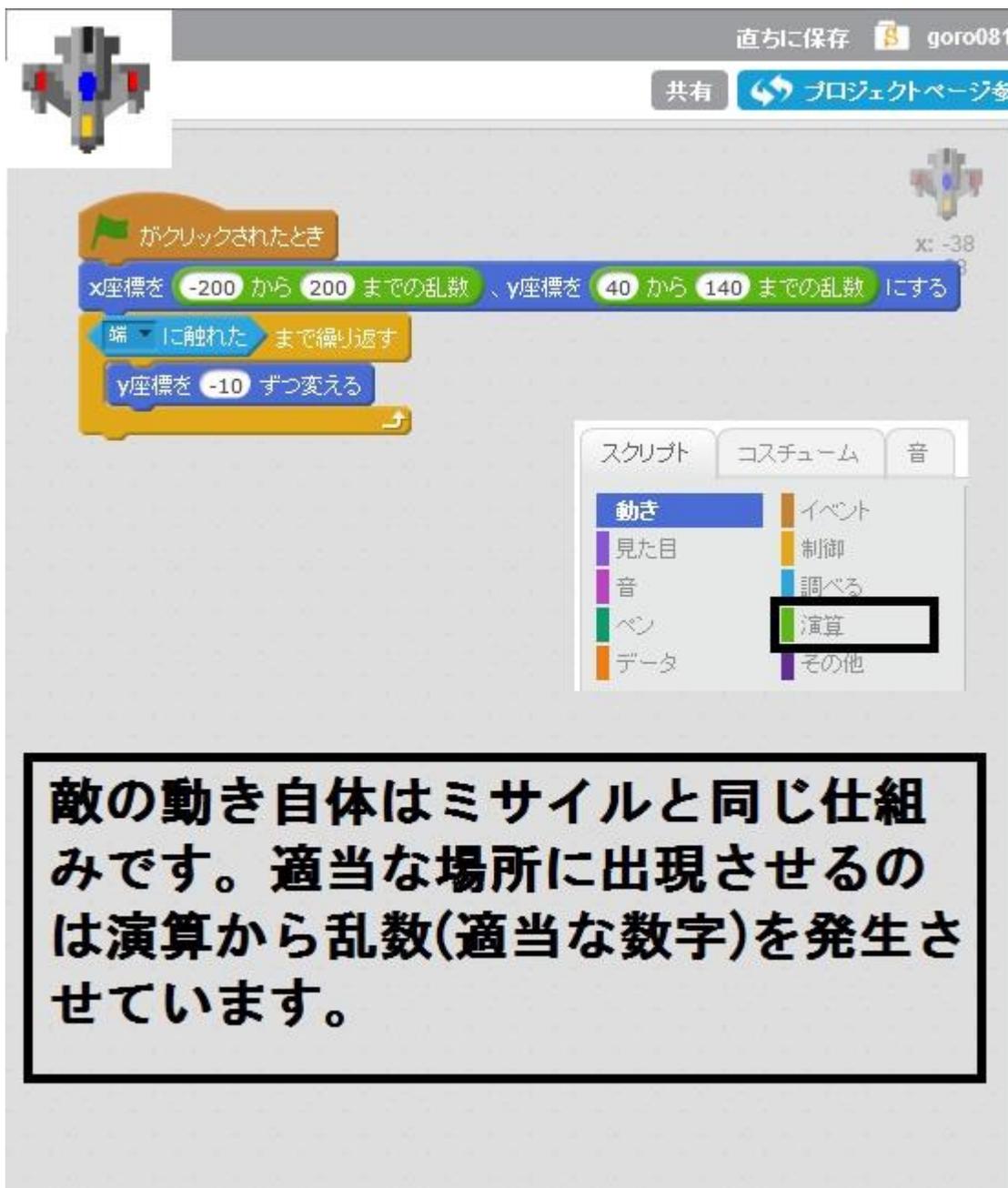
次に敵の動きを決めます。

敵の動きが予測できるとゲームとしては面白くないので、敵は乱数を使って適当な場所に出現するようにします。

このとき敵の画像の大きさが縦 70、横 70 程度なので、上下左右から敵の大きさの半分($70 \div 2 = 35$)程度をそれぞれ内側に来るようにします。そうすることで、敵が出現時に画面からはみ出ることがなくなります。そこで以下のようにプログラムしました。

x 軸方向は左右の ± 240 から 40(敵の飛行機の半分 $35 + 5$)を引いた ± 200 を出現範囲とし、y 軸方向は 0 より 40 上の 40 から 180 (一番上) $- 40 = 140$ を出現範囲としました。

またプログラムの一番上の緑の旗がクリックされたときというのは、ゲームで言うスタートボタンのことです。



直ちに保存 goro081

共有 プロジェクトページ参

がクリックされたとき

x座標を -200 から 200 までの乱数、y座標を 40 から 140 までの乱数 にする

端に離れたまで繰り返す

y座標を -10 ずつ変える

動き イベント
見た目 制御
音 調べる
ペン 演算
データ その他

敵の動き自体はミサイルと同じ仕組みです。適当な場所に出現させるのは演算から乱数(適当な数字)を発生させています。

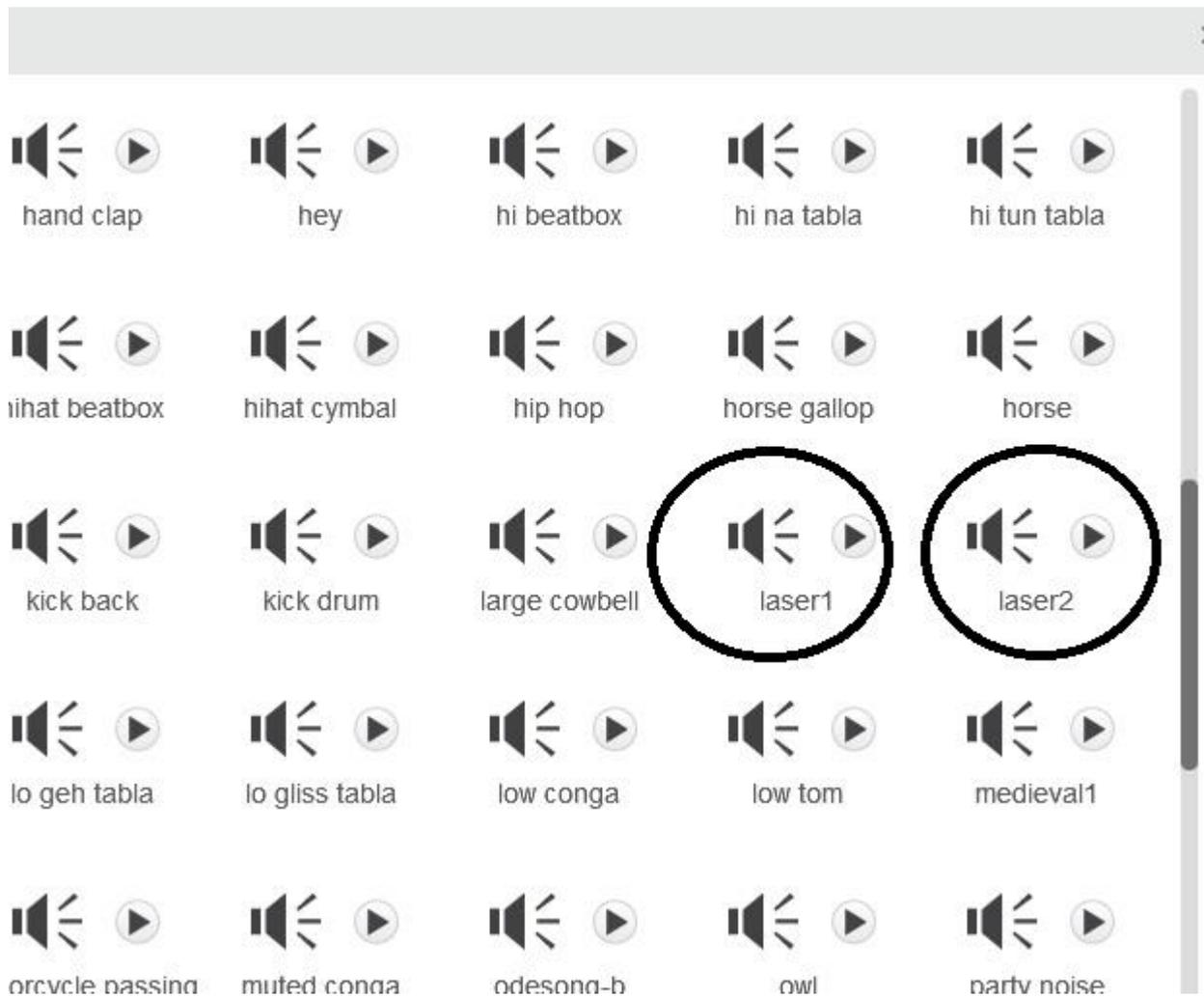
これで各キャラクターの大まかな動きが決まりました。



ゲームの音を追加します。
ミサイルの音とミサイルが敵に当たった時の音です。



leaser1(レーザー1)が発射音、leaser2(レーザー2)がミサイルが当たった時の音になります。



ミサイルのプログラムに laser1 を追加します。

同時にミサイルが移動中に敵に触れたらメッセージを送ることにします。これはミサイルが敵に当たったという合図を敵のプログラムに送ることを意味します。



The screenshot shows the Scratch script editor for a sprite named 'bomb'. The script is as follows:

- When the space key is pressed, play the sound 'laser1'.
- Show the sprite.
- Move the sprite 'playing111' to the top.
- Repeat the following loop until the end is reached:
 - Change the y-coordinate by 10.
 - If the sprite 'atchr1' is touched, send a 'Hit' message.
- Hide the sprite.

Below the script editor, there is a large text box with the following Japanese text:

**ミサイルにレーザー1の音を追加します。
ミサイルが端に行く間で敵に触れれば、ヒットというメッセージを敵に送るようにプログラムします。**

敵のプログラムを調整しましょう。

ミサイルから送られたヒットというメッセージを受け取ったなら、敵の飛行機を消してしまいうようにプログラムします。そのとき同時にミサイルに当たった音を鳴らします。

直ちに保存 goro081

共有 プロジェクトページを

atchr1

がクリックされたとき

ずっと

表示する

x座標を -200 から 200 までの乱数、y座標を 40 から 140 までの乱数 にする

端に到達するまで繰り返す

y座標を -2 ずつ変える

速さは -1~-3 で調整しましょう

ヒットを受け取ったとき

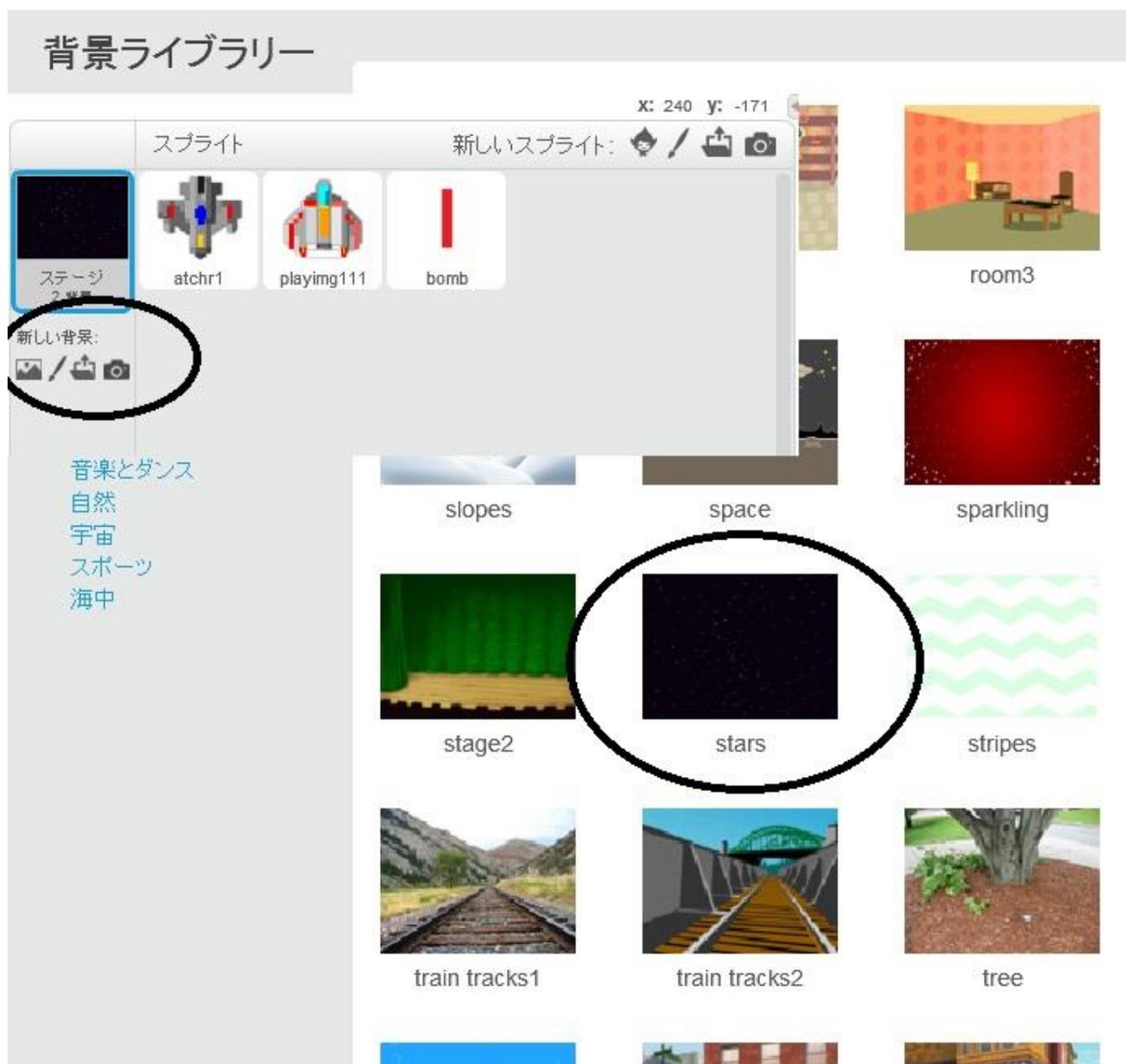
laser2 の音を鳴らす

隠す

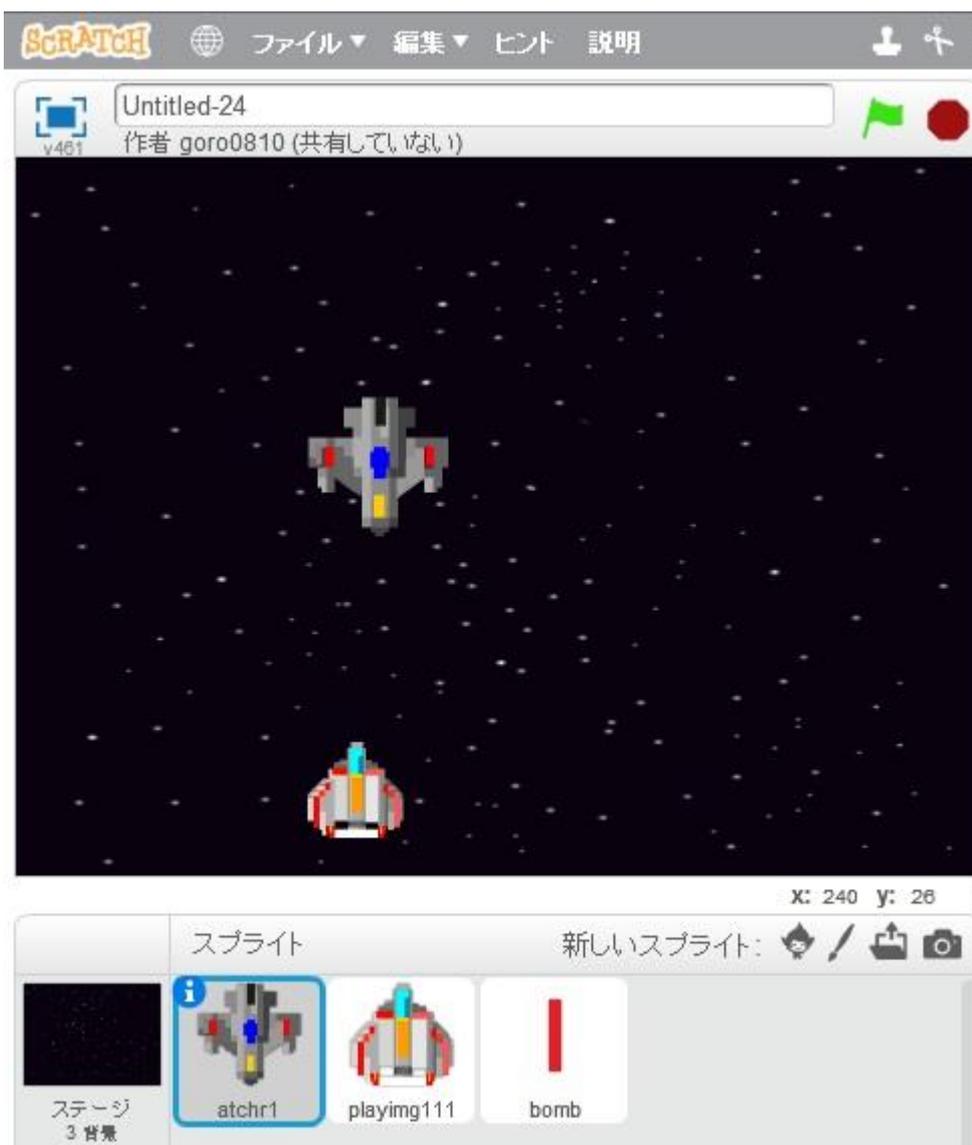
敵のほうではミサイルから送られてきたヒットを受け取るとレーザー2の音を鳴らして消えるようにプログラムします。

速さは-1~-3 としていますが、それ以外でも大丈夫です。速さに-(マイナス)がついているのは下に進むという意味です。速さの設定はゲームのバランスを大きく左右します。ゲームはバランスが大切です。みんなが楽しめる速さの設定を選んでくださいね。

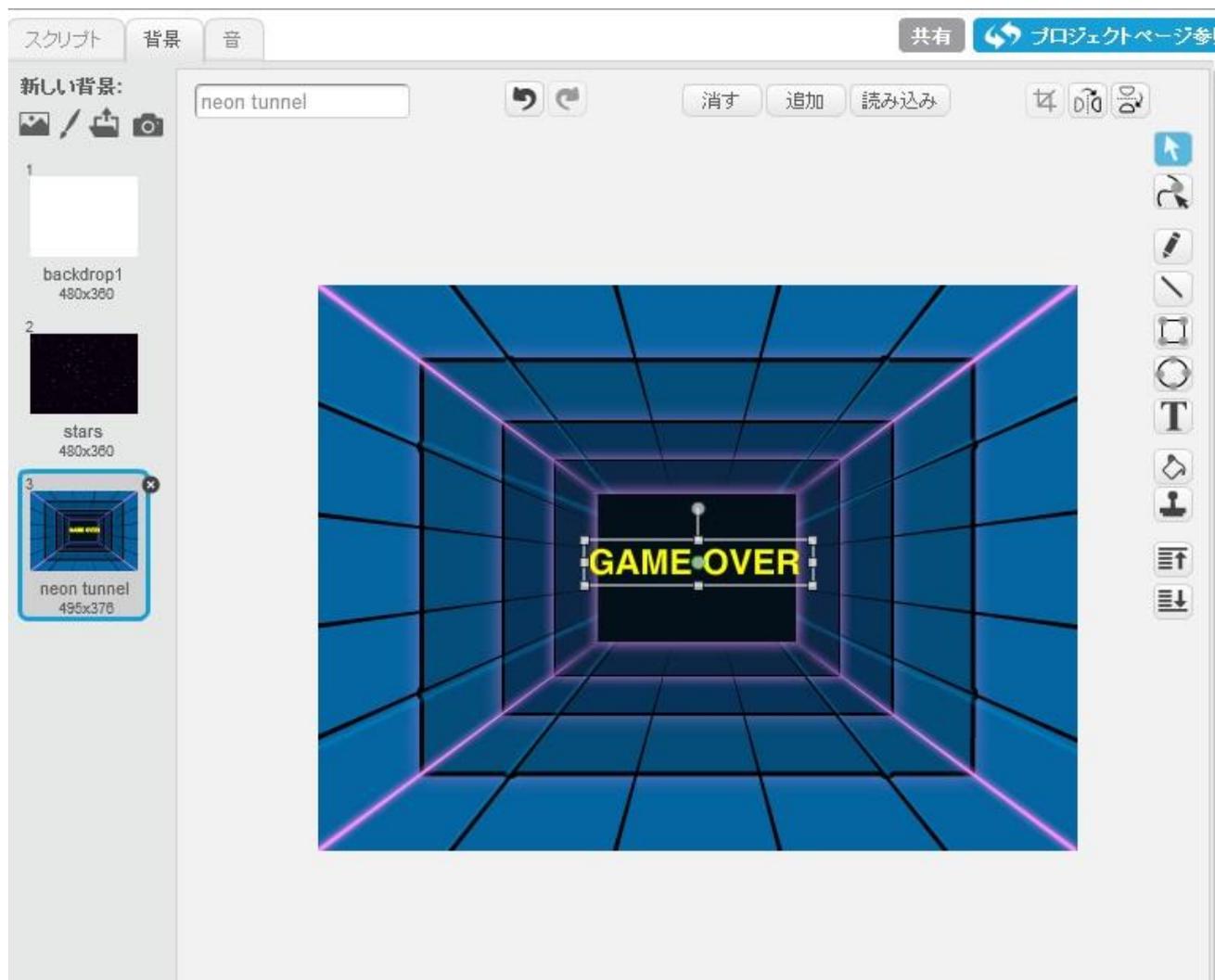
次に、背景を選択してシューティングゲームらしくしましょう。



星の背景に変更になりました。ゲームらしくなりましたね。



次にゲームオーバーの画面を作りましょう。
新しい背景を追加し、**GAME OVER** と書きましょう。
日本語は対応していないようです。



T が文字を書くとき選択するボタン
← は移動や拡大縮小のときに使います。

ゲームオーバーの条件として、敵が自分の飛行機に当たったらゲームオーバーとします。その当たり判定を下のように追加します。

The image shows a Scratch script for a game. The script starts with a 'がクリックされたとき' (when clicked) event, followed by '背景を stars にする' (set background to stars), 'ずっと' (forever) loop containing '表示する' (show), 'x座標を -200 から 200 までの乱数、y座標を 40 から 140 までの乱数 にする' (set x and y coordinates to random values), and a '端 に触れた' (when hit edge) block. A black circle highlights an 'もし' (if) block with the condition 'playimg111 に触れた なら' (if hit playimg111) and the action 'ゲームオーバー を送る' (send game over). To the right, there is a 'ヒット を受け取ったとき' (when hit) block with actions 'laser2 の音を鳴らす' (play laser2 sound) and '隠す' (hide). Below the script, two callout boxes provide explanations: one for the 'ゲームオーバー を受け取ったとき' (when game over received) block, stating 'ゲームオーバーを受け取ったら隠してプログラムを止めます' (when game over received, hide and stop program), and another for the 'もし' block, stating '敵が自分の飛行機に触れたらゲームオーバーとします。敵が自分の飛行機に触れたらゲームオーバーというメッセージを送ります。' (when enemy hits player's plane, game over. send game over message when enemy hits player's plane).

背景にもプログラムします。

敵が自分の飛行機に当たったらゲームオーバーのメッセージを送ります。

そのメッセージをこの背景が受け取ったら音を鳴らしてこのゲームオーバーの背景にします。そしてプログラムを停止させます。

スクリーンショットはScratchの「スクリプト」タブを示しています。左側の「ステージ」ウィンドウには「ゲームオーバー」メッセージが表示されています。中央の「新しい背景」リストには「backdrop1 480x360」、「stars 480x360」、および「neon tunnel 495x376」がリストアップされています。右側の「スクリプト」ウィンドウには、以下のスクリプトが記述されています：

- ゲームオーバーを受け取ったとき
- crash cymbal の音を鳴らす
- 背景を neon tunnel にする
- すべてを止める

また、右側の「新しい背景」リスト内の「neon tunnel」背景が黒い円で囲まれています。左下の大きな黒い枠には、以下の日本語が記述されています：

ゲームオーバーを受け取ったらこの背景を表示するようにプログラム。

ただこれだけだと、ゲームオーバーになってゲームを再開するときに、背景がゲームオーバーのままなので、次のようにゲームスタートをすると背景をゲームの画面にするようにプログラムします。

以下のようにゲーム開始ボタンが押されたら背景をゲーム中の背景にするようにプログラムします。注)このプログラムは背景に直接しても OK です。プログラムは算数や数学と同じように、作り方にはいろんな方法があります。



The image shows a Scratch script for a game. At the top, there is a toolbar with a user icon labeled 'atchr1', a '保存しました' (Saved) notification, and a 'goro081' profile. Below the toolbar are '共有' (Share) and 'プロジェクトページ' (Project Page) buttons. The main script is as follows:

- When clicked:** Set background to 'stars' (circled in red).
- When green flag clicked:**
 - Show
 - Set x coordinate to random number between -200 and 200, and y coordinate to random number between 40 and 140.
 - Repeat until edge is reached:
 - Change y coordinate by -2.
 - If 'playing111' is touched:
 - Send 'Game Over'.
- When 'Hit' is received:**
 - Play 'laser2' sound.
 - Hide.
- When 'Game Over' is received:**
 - Hide.
 - Stop everything.

A text box on the right side of the script reads: **ゲーム開始時はゲーム画面なのでその画面を表示させるようにする** (When the game starts, it's the game screen, so make sure to display that screen).

自分の飛行機もこのままだとゲームオーバーのとき、自分の飛行機が消えたあと、再びゲームを開始しても自分の飛行機が表示されないので、ゲーム開始ボタンが押されたら、自分の飛行機を表示するようにプログラムします。

The image shows a Scratch script editor with the following code blocks:

- When key pressed:** Right arrow key pressed → x coordinate increases by 10.
- When key pressed:** Left arrow key pressed → x coordinate decreases by 10.
- When game over is received:** Hide and stop all.
- When green flag is clicked:** Show.

Annotations and text boxes:

- A speech bubble points to the "When green flag is clicked" block with the text: **スタートボタンが押されたら表示させるようにします。**
- A box surrounds the "When game over is received" block with the text: **敵と同じように**

Other visible elements include a user profile for "playing111", a "共有" (Share) button, and a "プロジェクトページ" (Project page) link. The top right shows "保存しました" (Saved) and the user ID "goro0810".

ミサイルもゲームオーバーのときに消えるようにプログラムします。



The image shows a Scratch code editor window. At the top left, there is a sprite named 'bomb' represented by a red vertical bar. The main workspace contains a script for the 'bomb' sprite. The script starts with an event block 'スペース キーが押されたとき' (When space key is pressed). This is followed by a sequence of actions: 'laser1 の音を鳴らす' (Play sound laser1), '表示する' (Show), 'playing111 へ行く' (Go to playing111), and a loop '端 に触れた まで繰り返す' (Repeat until touching the end). Inside the loop, there is an action 'y座標を 10 ずつ変える' (Change y coordinate by 10) and a conditional 'もし atchr1 に触れた なら' (If touching atchr1). Inside this conditional is the action 'ヒット を送る' (Send hit). After the loop, there is an action '隠す' (Hide). Below this main script, a separate script is highlighted with a black box. It starts with an event block 'ゲームオーバー を受け取ったとき' (When game over is received), followed by '隠す' (Hide) and 'すべて を止める' (Stop all).

敵と同じように

これでゲームができるはずです。早速遊んでみてください。

これまでのプログラムのまとめを次のページに載せています。

自分の飛行機



敵の飛行機

The image shows a Scratch script for a character named "atchr1". The script is organized into several event-driven blocks:

- When clicked:** Sets the background to "stars".
- Forever loop:** Displays the character, sets its x-coordinate to a random number between -200 and 200, and its y-coordinate to a random number between 40 and 140. It then loops until it reaches the right edge, and increments its y-coordinate by -2.
- When touching "playing111":** Sends a "Game Over" message.
- When hit:** Plays the "laser2" sound and hides the character.
- When "Game Over" is received:** Hides the character and stops all scripts.

At the top right of the workspace, the character's current coordinates are shown as x: 111 and y: -128. The interface also includes a "Save" button, a "Share" button, and a "Project Page" link.

ミサイル

The image shows a Scratch script for an object named "bomb". The script is organized into two event-driven blocks:

- When Space Key is Pressed:**
 - Play sound `laser1`
 - Show
 - Go to `playing111`
 - Repeat until `end` is reached:
 - Change `y` coordinate by `10`
 - If `atchr1` is reached:
 - Send `hit`
 - Hide
- When Game Over is Received:**
 - Hide
 - Stop all

ゲームオーバーの背景



ゲームの難易度をあげたければ、複製して敵の数を増やしてみよう。



複製した敵は、少しプログラムを修正しないと、もとの敵と同じように動いてしまうので、以下のように修正しよう。複製すると atchr2 という敵の飛行機ができます。敵の飛行機とミサイルのプログラムを次のように修正します。

保存しました goro0810

共有 プロジェクトページ参照

atchr2

がクリックされたとき

背景を stars にする

ずっと

表示する

x座標を -200 から 200 までの乱数、y座標を 40 から 140 までの乱数 にする

端 に触れた まで繰り返す

y座標を -3 ずつ変える

もし plimg111 に触れた なら

ゲームオーバー を送る

ヒット2 を受け取ったとき

laser2 の音を鳴らす

隠す

ゲームオーバー を受け取ったとき

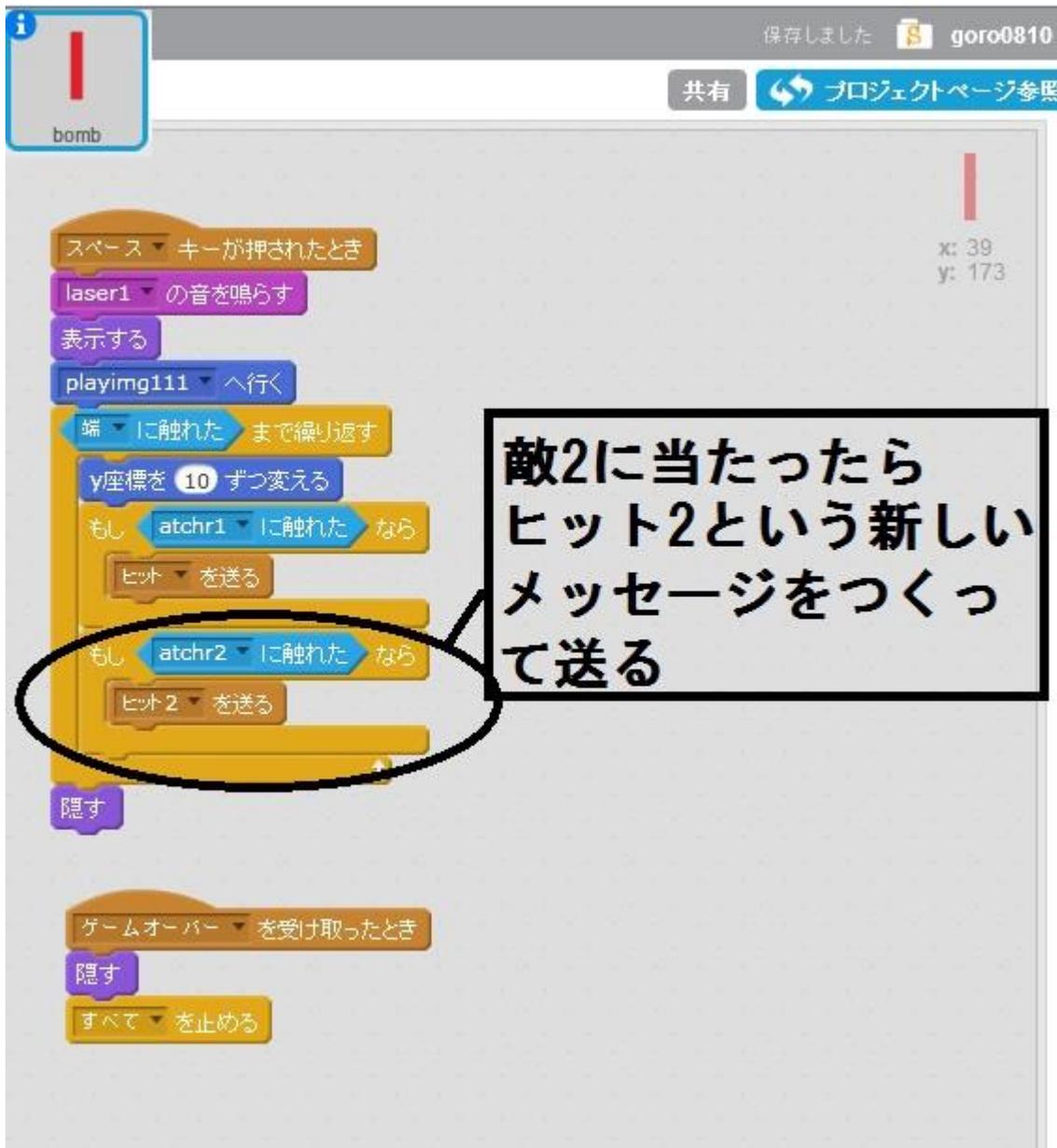
隠す

すべて を止める

x: 81
y: -116

- 3にして少し速くした

ヒット2という新しいメッセージをつくらせて当たり判定をする



その他、完成したプログラムをいじり倒してください。

そして楽しんでください。

敵からミサイルが出るプログラムなんて面白そうですね。

プログラミングの秘訣はトライアンドエラー(挑戦と失敗)の繰り返しですから。

今回ゲームのキャラクターはこちらで用意しましたが、もともとスクラッチにある画像を選んで同じプログラムを組めば、同じことができます。その時はキャラクターを縮小したりして適切なサイズにしてみてください。本もたくさん出ておりますので、そちらもご参考ください。